
Trabajo en Clase 1

Verificando el comportamiento de K Nearest Neighbors KNN

Fecha de asignación: 15 de Marzo, 2023
Grupos: 2 personas

Fecha de entrega: 24 de Marzo, 2023
Profesor: Jason Leitón Jiménez

1. Objetivo

Aplicar los conceptos básicos de instance learning para evaluar el comportamiento del algoritmo knn, utilizando el código visto en clase.

2. Indicaciones

1. Para el siguiente laboratorio deberá ejecutar los pasos que se detallan en cada uno de los apartados, comprobando su ejecución con capturas de pantalla en el resultado del mismo. Esta información deberá estar en un documento PDF para cargarlo en Tecdigital según corresponda.
2. Se debe de realizar la guía de preguntas y adjuntar las soluciones junto con enunciados en el mismo documento del punto anterior.
3. La fecha de entrega será la indicada en este documento y debe ser de manera individual, a menos de que se indique lo contrario.
4. En caso de que el laboratorio necesite código fuente, este también debe de incorporarse como parte de la solución del mismo, ya que será evaluado.
5. Todo procedimiento debe realizarse con algún editor de texto, por ejemplo Latex, word, markdown, entre otros.
6. El laboratorio debe ser revisado por el profesor antes de la fecha de cargar los archivos (en caso de que se indique) , por lo que el estudiante será el encargado de mostrar su trabajo, en caso de que no lo haga la nota será cero. En modalidad virtual este punto no aplica.

3. Teoría

1. Indique cual es costo de “Training” y de test del algoritmo KNN.
2. Explique porqué se dice que instance learning es un algoritmo por fuerza bruta o perezoso.

3. Indique cuantas comparaciones se deben realizar en la etapa de testeo (asumiendo el peor de los casos).
4. Investigue alguna modificación que se ha realizado al algoritmo original del KNN.

4. Realizando un clasificador de múltiples clases

Para esta sección se debe utilizar el código visto en clase, el cual se encuentra en tecdigital en un archivo llamado “knn.ipynb”.

4.1. Distancia entre elementos

1. Se debe de implementar un criterio de cómo elegir la clase de una muestra tomando en cuenta los k vecinos más cercanos. En el código se muestra el método normal (la mayoría de elementos de una misma clase es la que gana). La idea es que se implemente otro método tomando en cuenta la distancia entre los elementos. Este código debe implementarse dentro del método “predict” (en el if de distance). En el PDF se debe explicar un diagrama en qué consiste el algoritmo implementado. Es importante destacar que la métrica puede ser inventada por cada grupo
2. Pruebe el algoritmo del punto anterior con $k=2,3,4,5,6,7,9,11$. Anote la precisión y el tiempo que se obtuvo para cada k . Además, compare los resultados obtenidos de su implementación con los de la biblioteca **sklearn**.
3. Realice una gráfica de k vs precisión, de tal manera que le permita discutir los resultados obtenidos.

4.2. Clasificador de múltiples clase

En esta sección se requiere probar el mismo código de la sección anterior, sin embargo, se debe utilizar otro set de datos.

1. Cargue el set datos MNIST (dígitos), utilizando la biblioteca sklearn MNIST. En el listing 1 se muestra cómo hacerlo.

Listing 1: Cargar el Set de Datos

```
from sklearn.datasets import load_digits
import matplotlib.pyplot as plt
digits = load_digits(n_class=10)
```

2. Muestre las primeras 12 imágenes en forma de tabla (3x4). Se debe de observar la imagen del dígito.
3. Muestre las dimensiones del set datos (cuántos datos y cuántos features).
4. Separe el set de datos de tal manera que el 75 % de los datos sea para entrenamiento, mientras que el 25 % sea para el test.
5. Instancie su propia implementación de KNN y pruebe el clasificador ($k=2,3,5,7,8,10,11$) con el set datos solicitados (note que son 10 clases). Muestre la precisión con el método normal y con el método de distancia implementado por cada grupo.
6. Instancie el clasificador de sklearn y realice las mismas pruebas que el punto anterior.
7. Compare en una tabla la implementación propia del knn con el de la biblioteca sklearn.
8. Realice 5 imágenes con dígitos escrito a mano. Saque una foto a cada dígito y transforme la imagen a las mismas dimensiones que el set de datos. Pruebe su modelo prediciendo cada una de las fotos tomadas. Reporte la clasificación que brinda el modelo mostrando la imagen original y la clasificación obtenida.